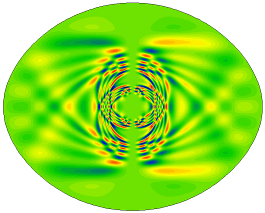
2D oscillation computations in rapidly rotating stars with the TOP code

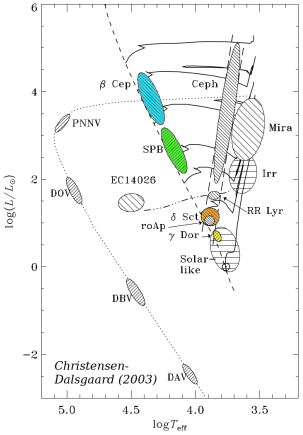
D. R. Reese

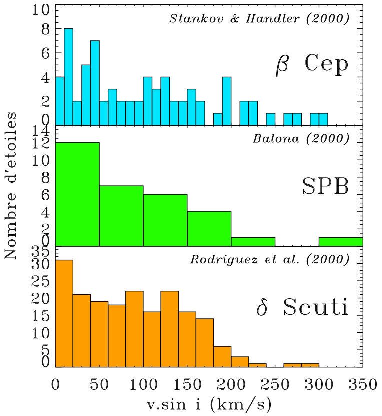
LESIA, Paris Observatory

November 29, 2018



Scientific context



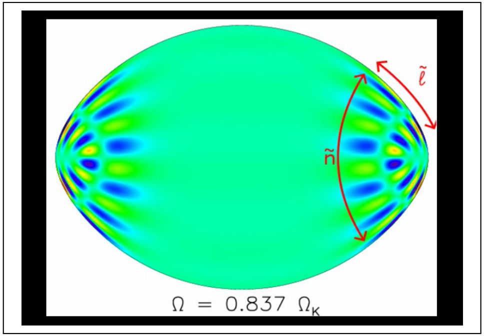


- seismic interpretation of rapidly rotating stars

Scientific context

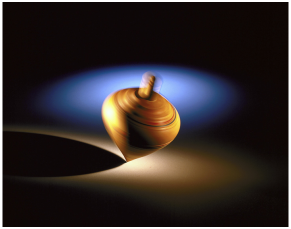
Effects of rapid rotation

- centrifugal deformation
- Coriolis force
- significant distortion of pulsation modes
- need for a 2D numerical approach



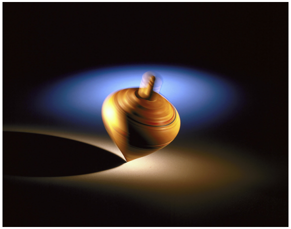
The TOP code

- TOP = Two-dimensional Oscillation Program
- top = "toupie" (in French)


http://johnmannophoto.com/blog/?p=103

The TOP code

- TOP = Two-dimensional Oscillation Program
- top = "toupie" (in French)

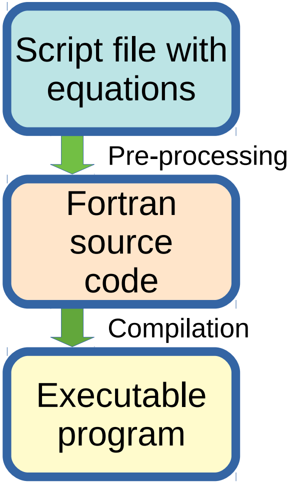

http://johnmannophoto.com/blog/?p=103

- initially started writing the code in 2007-2009 during first postdoc in Sheffield
    - numerical method based on Lignières et al. (2006)
    - borrowed parts from LSB (= Linear Solver-Builder)
- multi-domain version 2010-2011 (with J. Ballot)
- non-adiabatic version 2012-2013 (with M.-A. Dupret)
- B. Putigny unified multiple versions, added a python interface, and started redoing the script language (∼2013-present)

The TOP code

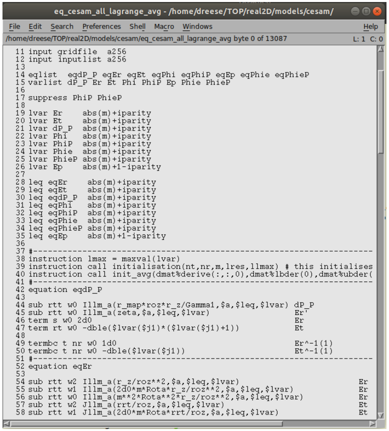
Basic characteristics

- Fortran 90 code
- Perl code for interpreting pulsation equations in script file
  - this produces Fortran code, prior to compilation
  - may be replaced by a newer interpreter in C++ & other language



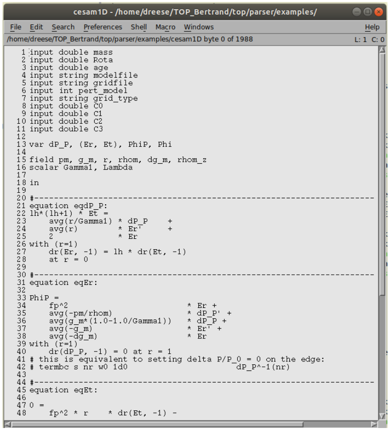
The TOP code

Current script


eq_cesam_all_lagrange_avg - /home/dreese/TOP/real2D/models/cesam/
File Edit Search Preferences Shell Macro Windows Help
/home/dreese/TOP/real2D/models/cesam/eq_cesam_all_lagrange_avg byte 0 of 13087 L: 1 C: 0

```
input gridfile  a256
input inputlist a256

eqlist  eqdP_P eqEr eqEt eqPhi eqPhiP eqEp eqPhie eqPhieP
varlist dP_P Er Et Phi PhiP Ep Phie PhieP

suppress PhiP PhieP

lvar Er    abs(m)+iparity
lvar Et    abs(m)+iparity
lvar dP_P  abs(m)+iparity
lvar Phi   abs(m)+iparity
lvar PhiP  abs(m)+iparity
lvar Phie  abs(m)+iparity
lvar PhieP abs(m)+iparity
lvar Ep    abs(m)+1-iparity

leq eqEr    abs(m)+iparity
leq eqEt    abs(m)+iparity
leq eqdP_P  abs(m)+iparity
leq eqPhi   abs(m)+iparity
leq eqPhiP  abs(m)+iparity
leq eqPhie  abs(m)+iparity
leq eqPhieP abs(m)+iparity
leq eqEp    abs(m)+1-iparity

#---------------------------------------------------------------------
instruction lmax = maxval(lvar)
instruction call initialisation(nt,nr,m,lres,llmax) # this initialises
instruction call init_avg(dmat%derive(:,:,0),dmat%lbder(0),dmat%ubder(
#---------------------------------------------------------------------
equation eqdP_P

sub rtt w0 Illm_a(r_map*roz*r_z/Gamma1,$a,$leq,$lvar) dP_P
sub rtt w0 Illm_a(zeta,$a,$leq,$lvar)                 Er'
term s w0 2d0                                         Er
term rt w0 -dble($lvar($j1)*($lvar($j1)+1))           Et

termbc t nr w0 1d0                                    Er^-1(1)
termbc t nr w0 -dble($lvar($j1))                      Et^-1(1)
#---------------------------------------------------------------------
equation eqEr

sub rtt w2 Illm_a(r_z/roz**2,$a,$leq,$lvar)                        Er
sub rtt w1 Illm_a(2d0*m*Rota*r_z/roz**2,$a,$leq,$lvar)             Er
sub rtt w0 Illm_a(m**2*Rota**2*r_z/roz**2,$a,$leq,$lvar)           Er
sub rtt w2 Jllm_a(rrt/roz,$a,$leq,$lvar)                           Et
sub rtt w1 Jllm_a(2d0*m*Rota*rrt/roz,$a,$leq,$lvar)                Et
```

New script


cesam1D - /home/dreese/TOP_Bertrand/top/parser/examples/
File Edit Search Preferences Shell Macro Windows Help
/home/dreese/TOP_Bertrand/top/parser/examples/cesam1D byte 0 of 1988 L: 1 C: 0

```
input double mass
input double Rota
input double age
input string modelfile
input string gridfile
input int pert_model
input string grid_type
input double C0
input double C1
input double C2
input double C3

var dP_P, (Er, Et), PhiP, Phi

field pm, g_m, r, rhom, dg_m, rhom_z
scalar Gamma1, Lambda

in

#---------------------------------------------------------------------
equation eqdP_P:
lh*(lh+1) * Et =
    avg(r/Gamma1) * dP_P    +
    avg(r)        * Er'     +
    2             * Er
with (r=1)
    dr(Er, -1) = lh * dr(Et, -1)
    at r = 0

#---------------------------------------------------------------------
equation eqEr:

PhiP =
    fp^2                         * Er +
    avg(-pm/rhom)                * dP_P' +
    avg(g_m*(1.0-1.0/Gamma1))    * dP_P +
    avg(-g_m)                    * Er' +
    avg(-dg_m)                   * Er
with (r=1)
    dr(dP_P, -1) = 0 at r = 1
# this is equivalent to setting delta P/P_0 = 0 on the edge:
# termbc s nr w0 1d0                         dP_P^-1(nr)

#---------------------------------------------------------------------
equation eqEt:

0 =
    fp^2 * r    * dr(Et, -1) -
```

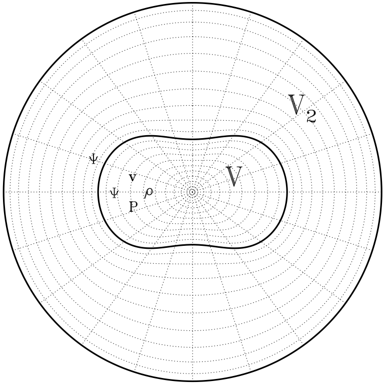
The TOP code

Numerical aspects and parallelisation

- TOP relies on BLAS and LAPACK for most of the heavy computations
  - good optimisation on most (super-)computers
  - provides OpenMP type parallelisation
- experimented with ScaLAPACK library
  - provides MPI type parallelisation
  - may allow solving 3D problems (by increasing available RAM)

Numerical approach

- 2D numerical approach
- Angular discretisation: spherical harmonic
- Radial discretisation:
  - polytrope: Chebyshev
  - realistic model: FD or splines
- Uses a suitable coordinate system (ζ, θ, ϕ) (cf. Bonazzola et al., 1998)



$$[\text{System of equations}] \Rightarrow Av = \lambda Bv$$

Method

1. Find explicit equations in spheroidal coordinates

Continuity equation
Euler's equation
Adiabatic relation
Poisson's equation

Method

1. Find explicit equations in spheroidal coordinates

2. Express unknowns using spherical harmonics

For example: $$\Phi(\zeta,\theta,\varphi) = \sum_{\ell=|m|}^{\ell_{max}} \Phi_m^\ell(\zeta) Y_\ell^m(\theta,\varphi)$$

Method

1. Find explicit equations in spheroidal coordinates

3. Project equations on spherical harmonic basis

2. Express unknowns using spherical harmonics

For example: $$\iint_{4\pi} \left\{Y_\ell^m\right\}^* \{\text{Continuity equation}\}\, d\Omega$$

Method

1. Find explicit equations in spheroidal coordinates

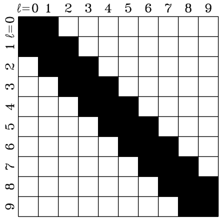
2. Express unknowns using spherical harmonics

3. Project equations on spherical harmonic basis

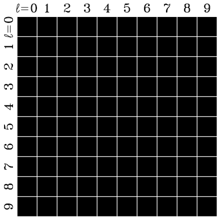
4. Carry out radial discretisation (Chebyshev, FD, splines)

Generalised eigenvalue problem: $Av = \lambda Bv$

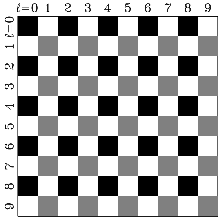
Horizontal discretisation



Only Coriolis



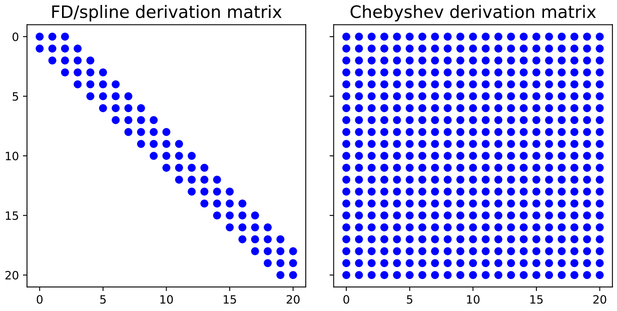
Centrifugal def.



Equat. symmetry

- spectral convergence with spherical harmonics
- problem couples all spherical harmonics of same parity

Radial discretisation



Discretisation	Convergence	Grid
FD/splines	algebraic	flexible
Chebyshev polynomials	exponential	fixed

Radial discretisation – Chebyshev polynomials

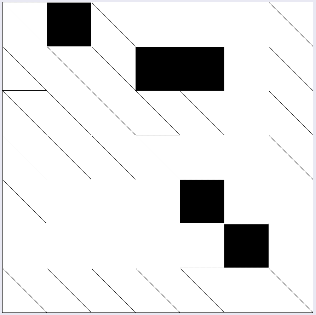
Characteristics

- construct & factorise full matrix
    - good parallelisation

Illustration

- **Model**: polytrope
- **Resolution**: 5670×5670
    - $N_r = 81$
    - $N_\theta = 10$
- **Fill factor**: $10.6\,\%$

Matrix



Radial discretisation – Chebyshev polynomials

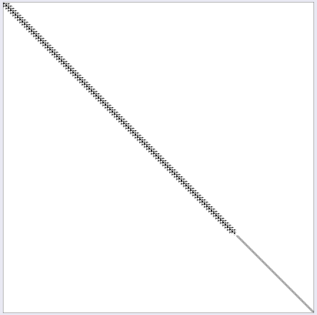
Characteristics

- construct & factorise band matrix
  - poor parallelisation

Illustration

- **Model**: SCF
- **Resolution**: 8080×8080
  - $(N_r, N_\theta) = (101, 10)$
  - Lower bands: 130
  - Upper bands: 140
- **Fill factor**: 27.0 %

Matrix



Matrix construction

- the order of the equations and the variables are set by the user
  - when dealing with FD/splines, group ℓ values together to obtain banded matrix
  - fine-tune ordering to reduce number of bands

Solving the eigenvalue problem

Arnoldi-Chebyshev algorithm

- iterative procedure in which original matrix is approximated by a smaller matrix obtained by successive iterations of: Av starting with some initial vector v_0

Solving the eigenvalue problem

Arnoldi-Chebyshev algorithm

- iterative procedure in which original matrix is approximated by a smaller matrix obtained by successive iterations of: Av starting with some initial vector v_0

Spectral transformation

$$Av = \lambda Bv \;\Leftrightarrow\; (A - \sigma B)^{-1} Bv = \mu v \qquad \text{where} \qquad \lambda = \sigma + \frac{1}{\mu}$$

- when targeting eigenvalues close to σ, we need to solve $(A - \sigma B)\, X = Y$
- it is therefore necessary to construct and factorise $A - \sigma B$

Polynomial eigenvalue problem

$$\sum_{i=0}^{n} \lambda^i A_i v = 0 \qquad \Longleftrightarrow \qquad \mathcal{A}x = \lambda \mathcal{B} x \qquad \text{where}$$

$$\mathcal{A} = \begin{bmatrix} A_0 & . & . & . \\ . & I_\mathrm{d} & . & . \\ & & \ddots & \\ . & . & & . \\ . & . & . & I_\mathrm{d} \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} -A_1 & -A_2 & \cdots & -A_n \\ I_\mathrm{d} & . & . & . \\ & \ddots & & \\ . & & . & . \\ . & . & I_\mathrm{d} & . \end{bmatrix}, \quad x = \begin{bmatrix} v \\ \lambda v \\ \vdots \\ \lambda^{n-1} v \end{bmatrix}$$

Solving $(\mathcal{A} - \sigma\mathcal{B})\,X = Y$

1. $X = [x_0 \ldots x_{n-1}]^T, \qquad Y = [y_0 \ldots y_{n-1}]^T$
2. By induction, let us define $(w_i)_{i \in [1, n-1]}$:
   $w_1 = \sigma y_1, \qquad w_{i+1} = \sigma(y_{i+1} + w_i)$
3. Solve: $x_0 = \left(\sum_{i=0}^{n} \sigma^i A_i\right)^{-1} \left(y_0 - \sum_{i=1}^{n-1} A_{i+1} w_i\right)$
4. By induction: $x_{i+1} = y_{i+1} + \sigma x_i$

The multi-domain approach

- **assumption**: only consecutive domains are coupled
  $\Rightarrow$ tridiagonal block matrix

$$\begin{bmatrix} A_{11} & A_{12} & & \\ A_{21} & A_{22} & A_{23} & \\ & & \ddots & A_{n-1,\,n} \\ & & A_{n,\,n-1} & A_{n,\,n} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}$$

Solving this system

- use of Gauss' pivot to eliminate $A_{i+1,\,i}$ and $A_{i,\,i+1}$
- one should not forget that matrix multiplication is not commutative

"Factorisation"

$$\tilde{A}_{11} = A_{11} \qquad \tilde{A}_{i+1,\,i+1} = A_{i+1,\,i+1} - A_{i+1,\,i}\tilde{A}_{i,\,i}^{-1}A_{i,\,i+1}$$

Downward sweep

$$\tilde{Y}_1 = Y_1 \qquad \tilde{Y}_{i+1} = Y_{i+1} - A_{i+1,\,i}\tilde{A}_{i,\,i}^{-1}\tilde{Y}_i$$

Upward sweep

$$X_n = \tilde{A}_{n,\,n}^{-1}\tilde{Y}_n \qquad X_{i-1} = \tilde{A}_{i-1,\,i-1}^{-1}\left(\tilde{Y}_{i-1} - A_{i-1,\,i}\tilde{X}_i\right)$$

Radial discretisation – spectral multi-domain

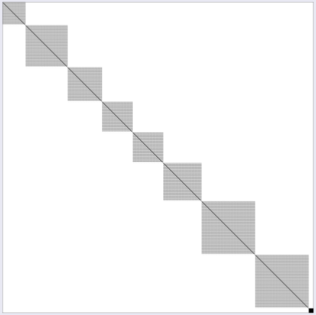
Characteristics

- tridiagonal block matrix
  - good parallelisation

Illustration

- **Model**: ESTER
- **Resolution**: 10150×10150
  - $N_r = (30, 55, 45, 40, 40, 50, 70, 70, 30)$
  - $N_\theta = 5$
- **Fill factor**: $25.4\,\%$

Matrix



Typical numerical resolutions

- polytropes: $N_r = 60$, $N_\theta = 40$
- SCF: $N_r = 1601$, $N_\theta = 10$ to 50
- **Note**: $\ell_{\max} \simeq 2N_\theta$

Numerical cost for adiabatic calculations in ESTER models

N_r	N_θ	Memory (in Gb)	Time (in min)	Num. proc.
400	10	0.5	0.16	2
400	15	1.1	0.33	2
400	20	1.9	0.65	2
400	30	4.2	1.6	2
400	40	7.4	3.3	2
400	100	$\sim$70	24	25

Numerical cost for non-adiabatic calculations in ESTER models

N_r	N_θ	Memory (in Gb)	Time (in min)	Num. proc.
400	10	3.5		
400	15	7.9		
400	20	13.4	5	4
400	29	28.0	10	8
400	40	52.7	22	8
400	50	82.3	26	16

Tests and precision of the method

Polytrope

- Comparison with Christensen-Dalsgaard and Mullan (1994) ($\Omega = 0$): $\Delta\omega/\omega \sim 10^{-7}$
- Comparison with Lignières et al. (2006): $\Delta\omega/\omega \sim 10^{-7}$
- Comparison with Saio (1981) for small Ω
- Variational principle: $\Delta\omega/\omega \sim 10^{-7}$ when $N = 3$ and $\Delta\omega/\omega \sim 10^{-5}$ when $N = 1.5$
- Numerically: $\Delta\omega/\omega \gtrsim 10^{-10}$
- used to validate the ACOR, another 2D pulsation code (Ouazzani et al. 2012)

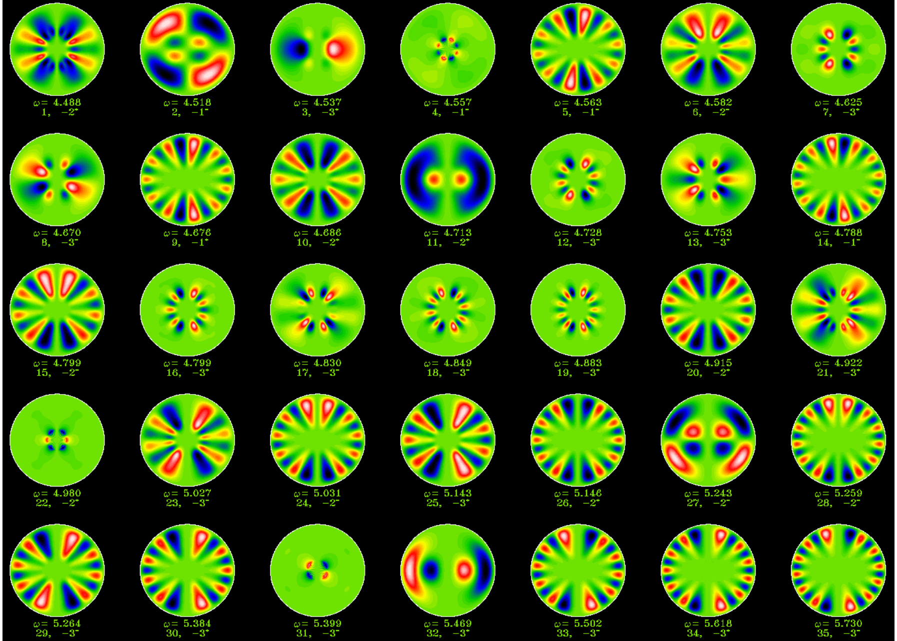
SCF

- Variational principle: $\Delta\omega/\omega \sim 10^{-4}$ à 10^{-3}
- Numerically: $\Delta\omega/\omega \sim 10^{-5}$ à 10^{-4}

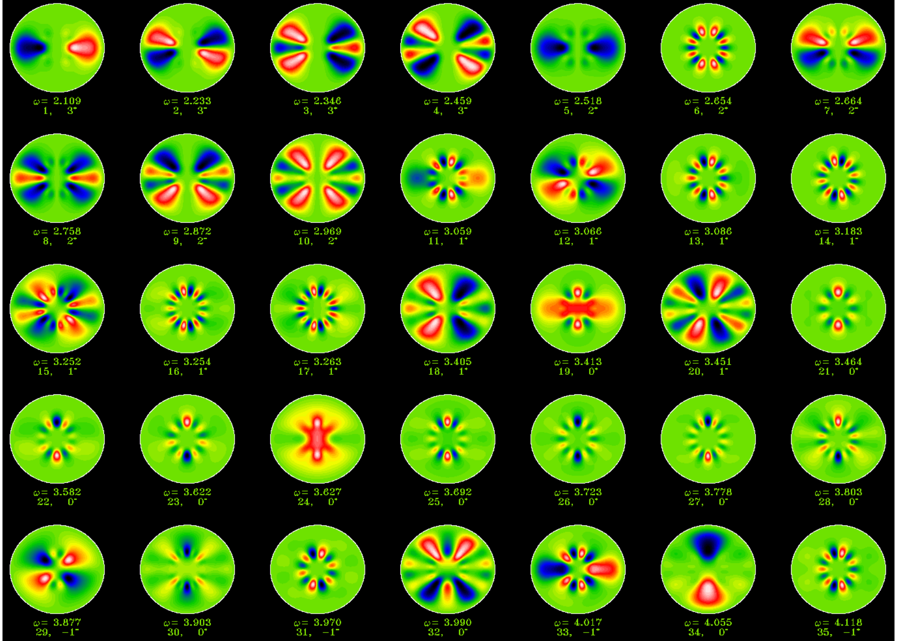
Tests and precision of the method – ESTER models

Estimated accuracy

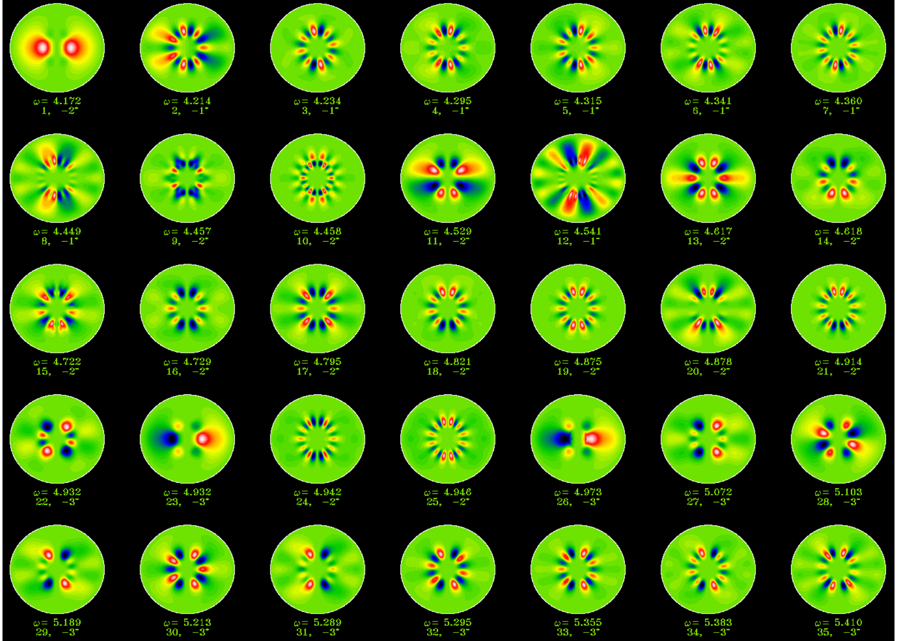
- frequencies:
  - analytical EOS: 10^{-8} to 10^{-7}
  - tabulated EOS: $\sim 10^{-4}$
- excitation/damping rates: 10^{-2} to 10^{-1}



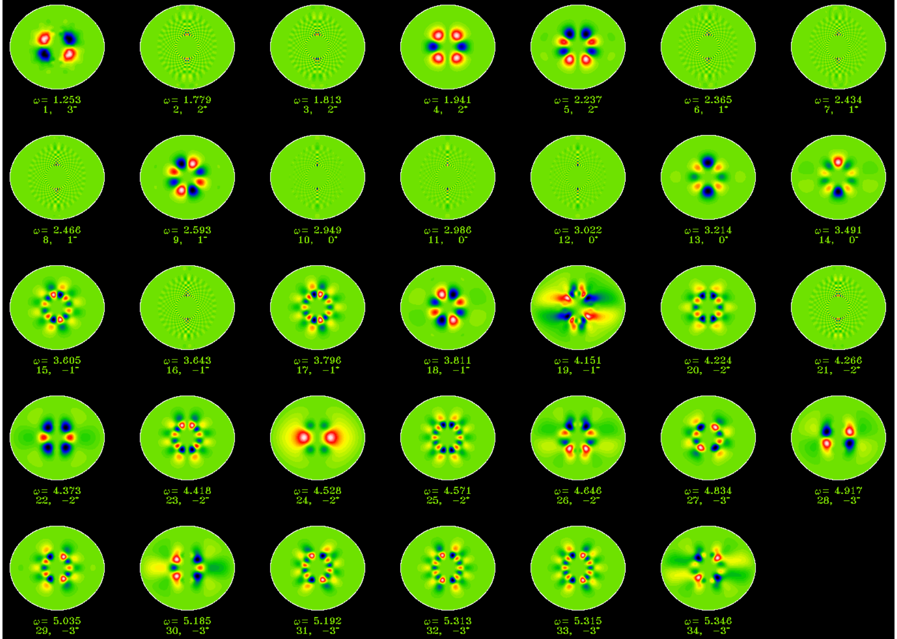
$\Omega = 0.4\ \Omega k$



$\Omega = 0.5\ \Omega k$

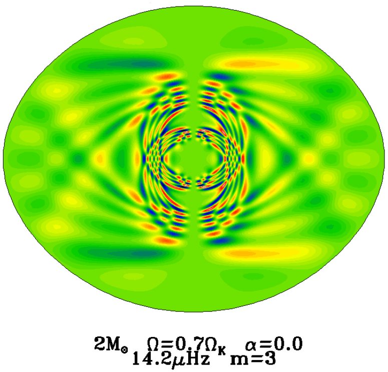


$\Omega = 0.5\ \Omega k$



$\Omega = 0.6\ \Omega k$

Rosette modes



(Reese, 2013)

- first discovered by Ballot et al. (2011) using the TOP code
- further studied by Takata & Saio (2013, 2014, 2014b)